

Express Mail Label N .: EL 979045323 US
Dat Mail d: Novemb r 24, 2003

UNITED STATES PATENT APPLICATION FOR GRANT OF LETTERS PATENT

**TRACY FULGHUM
GREGORY E. BOTTOMLEY
Y. -P. ERIC WANG
ALI S. KHAYRALLAH**

INVENTORS

METHOD AND APPARATUS FOR DS-CDMA INTERFERENCE SUPPRESSION USING CODE-SPECIFIC COMBINING

COATS & BENNETT, P.L.L.C.
P.O. Box 5
Raleigh, NC 27602
(919) 854-1844

METHOD AND APPARATUS FOR DS-CDMA INTERFERENCE SUPPRESSION USING
CODE-SPECIFIC COMBINING

BACKGROUND OF THE INVENTION

5

The present invention relates generally to RAKE receivers and particularly to interference suppressing RAKE receivers.

RAKE receivers represent a well-known approach to multi-path reception, particularly in Direct Sequence Code Division Multiple Access (DS-CDMA) wireless communication. In a multi-path communication channel, a transmitted signal travels through multiple propagation paths to the receiver. Thus, the receiver receives multiple “echoes” of the transmitted signal, with each multi-path echo generally suffering from different path delay, phase, and attenuation effects.

RAKE receivers exploit multi-path propagation by assigning each of two or more RAKE “fingers” to one of the incoming multi-path echoes. Each finger is tuned to a particular multi-path echo. By estimating the channel effects, e.g., phase and attenuation, and by properly accounting for the differences in path delays, the individual output from each RAKE finger may be combined with the outputs from the other fingers to provide a combined RAKE output signal with a greatly improved signal-to-noise ratio (SNR).

20 In DS-CDMA systems, such as Wideband CDMA and IS-2000, high transmission data rates are achieved by transmitting data at a low spreading factor and/or on more than one spreading code (multi-code). When a low spreading factor and/or multi-code is used, performance is sensitive to multi-path dispersion. With dispersion, there are multiple echoes of the transmitted signal with different relative delays. These echoes interfere with one another. Not only is orthogonality lost between successive symbols as one symbol overlaps with the next, but orthogonality is also lost between symbols sent on different, orthogonal codes.

25

As a result, performance is often limited by interference between different symbols being sent to a particular user. These symbols can correspond to the previous or next symbol, symbols sent in parallel on another code carrier, or both. In general, this interference is referred to as self-interference, where self-interference may include intersymbol interference (ISI) or inter-code interference (ICI). A key aspect of ISI is that it varies from symbol to symbol. This variation is due to the spreading codes being a combination of Walsh codes and a common scrambling code. The scrambling code has a period much longer than the symbol period, which effectively changes the overall spreading code from symbol period to symbol period.

SUMMARY OF THE INVENTION

The present application discloses a method and apparatus for reducing interference, such as inter-symbol interference, from a symbol of interest in a RAKE receiver. The RAKE receiver comprises a plurality of RAKE fingers, a processor, and a combiner. The plurality of RAKE fingers despread symbols received over multiple paths of a multi-path channel. The processor determines cross-correlations between different symbols. The combiner combines the despread symbols using the cross-correlations to reduce interference from the symbol of interest.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 comprises a block diagram of a conventional generalized RAKE (G-RAKE) receiver.

Figure 2 comprises a block diagram of an exemplary multi-symbol RAKE (M-RAKE) receiver according to the present invention.

Figure 3 comprises a block diagram of an exemplary restricted M-RAKE receiver according to the present invention.

Figure 4 illustrates an exemplary multi-channel linear transversal filter for the restricted M-RAKE of Figure 3.

Figure 5 illustrates an exemplary linear transversal filter for the multi-channel linear transversal filter of Figure 4.

5 Figure 6 illustrates an exemplary linear equalizer according to the present invention.

Figure 7 illustrates an exemplary multi-code linear equalizer according to the present invention.

Figure 8 illustrates an exemplary multi-channel filter for the multi-channel linear equalizer of Figure 7.

Figures 9A and 9B illustrate a mathematical technique for simplifying weight calculations associated with RAKE receivers.

Figure 10 illustrates an exemplary wireless communication system that utilizes the RAKE receivers of the present invention.

15 DETAILED DESCRIPTION OF THE INVENTION

Figure 1 illustrates a conventional wireless communication receiver 100 utilizing a generalized-RAKE (G-RAKE) 112. Wireless communication receiver 100 comprises receiver front end 110 and G-RAKE 112. The receiver 100 and/or the G-RAKE 112 may be embodied in an application specific integration circuit (ASIC). Receiver front end 110 outputs received signals $r(t)$ to G-RAKE 112. These received signals $r(t)$ comprise streams of sample values obtained from wireless signals received by one or more antennas associated with one or more receiver front ends 110. An exemplary front end 110 may include amplifiers, filters, mixers, digitizers, and/or other electronics as needed to produce a sampled signal suitable for processing by the G-RAKE 112. The received signal $r(t)$ output to G-RAKE 112 typically comprises a composite signal that includes one or more multi-path

echoes arising from multi-path propagation and which arrive at the receiver 100 from different directions and with different time delays. Further, received signal $r(t)$ may include components transmitted from or received by different antennas (not shown). The task of the G-RAKE 112 is to generate an estimate $\hat{s}(m)$ of each transmitted symbol $s(m)$ from the
5 received signal $r(t)$.

G-RAKE 112 comprises a plurality of RAKE fingers 120, RAKE processor 130, and RAKE combiner 132. Each RAKE finger 120 processes different time shifts or multi-path echoes of the received signal $r(t)$. Typically, each RAKE finger 120 comprises a delay element 122 and a correlator 124. Delay elements 122 delay the received signal $r(t)$ to time
10 align the multi-path echoes processed by each RAKE finger 120. Correlators 124 correlate the delayed signals with a spreading code to extract the assigned multi-path echoes from the received signal $r(t)$. Despread values from correlators 124 are combined in combiner 132. Combiner 132 typically includes weighting elements 126 and summer 134. Weighting elements 126 weight the multi-path echoes output from respective correlators 124. The
15 weighted multi-path echoes are summed symbol-by-symbol by summer 134 to form an estimate of a symbol of interest, $\hat{s}(m)$, during each symbol period. While some of the figures refer to the estimated symbol of interest as $\hat{s}(m)$, throughout this specification the symbol of interest $s(m)$ and the estimated symbol of interest $\hat{s}(m)$ may be represented by $s(0)$ and $\hat{s}(0)$, respectively, where $m=0$, without loss of generality. Further, those skilled in the art will
20 appreciate that the time shifts associated with delay elements 122 correspond to the delays of the multi-path echoes, while the combining weights associated with weighting elements 126 may correspond to the conjugates of the multiplying coefficients of the multi-path echoes (RAKE) or conjugates of weights that depend on the coefficients and a noise correlation matrix (G-RAKE).

In the conventional receiver 100 described above, the symbol estimates $\hat{s}(m)$ output during each symbol period may be corrupted by interference, such as intersymbol interference (ISI), which may result in demodulation and decoding errors. The ISI may be due in part to the dispersive effects of the communication channel, and in part to cross-correlations between the spreading codes used for different symbols. It should be noted that the spreading code is a combination of an assigned Walsh code and a scrambling code. The period of the scrambling code is typically larger than a symbol period, so the net effect is a time-varying spreading code. The present invention uses knowledge of the communication channel and cross-correlations between spreading codes for different symbols to determine cross-correlations between symbols to reduce interference.

Figure 2 illustrates an exemplary multi-symbol RAKE (M-RAKE) 200 according to the present invention. As with the G-RAKE 112, M-RAKE 200 may be embodied in an ASIC. M-RAKE 200 comprises a plurality of RAKE fingers 220, a RAKE processor 230 and a RAKE combiner 232. Unlike the RAKE fingers 120 of G-RAKE 112, RAKE fingers 220 may despread symbols from different symbol periods. Typically, one or more RAKE fingers 220 will have a delay that corresponds to a symbol of interest $s(0)$. Other RAKE fingers 220 will have delays that correspond to interfering symbols (i.e. symbols near the symbol of interest that contribute to intersymbol interference). During each symbol period, weighting elements 226 weight the despread symbols output from corresponding correlators 224 to form weighted outputs, which are then summed in summer 234 to form a symbol estimate $\hat{s}(0)$. RAKE processor 230 calculates the combining weights of weighting elements 226 to cancel or reduce intersymbol interference due to the effects of the communication channel and cross-correlations between spreading codes for different symbols.

The combining weights or weighting factors for M-RAKE 200 may be determined from a maximum likelihood (ML) or a minimum mean square error (MMSE) solution

computed by RAKE processor 230 utilizing knowledge of the different symbol spreading sequences and the channel coefficients. Typically, these combining weights are based on an analysis of the correlator outputs. With many of the mathematical details and definitions deferred till later, the vector of correlator outputs may be represented as:

$$\begin{aligned} \mathbf{y} &= [y(d_1, m_1) \ y(d_2, m_2) \ \dots \ y(d_J, m_J)]^T \\ &= \mathbf{h}_0 s(0) + \sum_{i \neq 0} \mathbf{h}_i s(i) + \mathbf{n}, \end{aligned} \quad (1)$$

where $y(d_j, m_j)$ corresponds to the output of delay element J and correlator J,

$\mathbf{h}_0 s(0) + \sum_{i \neq 0} \mathbf{h}_i s(i)$ represents the vector output components associated with each transmitted

symbol, and \mathbf{n} represents a noise output vector (thermal noise, other interference). Generally

speaking \mathbf{h}_i comprises the composite channel impulse response associated with the i^{th} symbol, $s(i)$, which includes the channel effects and despreading (spreading sequence cross-correlations). The first term in Eq. (1) represents the symbol of interest and the summation in Equation 1 represents the contribution of all other symbols, or the ISI, in \mathbf{y} .

The symbol of interest, $s(0)$, is estimated by using a linear combination of the outputs of RAKE fingers 220, which may be represented as $\hat{s}(0) = \mathbf{w}^H \mathbf{y}$, where \mathbf{w}^H is the conjugate transpose of a weight solution vector \mathbf{w} . The linear combination is performed in RAKE combiner 232 by weighting elements 226 and summer 234.

The weight solution vector, \mathbf{w} , based on a maximum likelihood criterion for the estimate, may be calculated by:

$$\mathbf{w} = \mathbf{R}_u^{-1} \mathbf{h}_0, \quad (2)$$

where \mathbf{R}_u is the correlation matrix of all undesired components in Equation 1. Correlation matrix \mathbf{R}_u may be calculated by taking the expectation with respect to noise and random transmitted symbols (assuming the random transmitted symbols are independent). The resulting correlation matrix of the undesired components may be represented by:

$$\mathbf{R}_u = \sum_{i \neq 0} \mathbf{h}_i \mathbf{h}_i^H + \mathbf{R}_n, \quad (3)$$

where \mathbf{R}_n is the correlation matrix of the noise, which may be correlated across fingers after the despreading process. The calculation of the noise correlation matrix \mathbf{R}_n is shown below.

Because the expectation is taken over random transmitted symbols, instead of the known

5 spreading codes, the combining weights applied by weighting elements 226 will change each symbol period due to the time-varying nature of the spreading waveform.

The derivation of the correlator outputs defined in Equation 1 will now be described.

The transmitted signal for a single code channel may be defined as:

$$x(t) = \sqrt{E} \sum_{i=-\infty}^{\infty} s(i) a_i(t - iT), \quad (4)$$

10 where E is the symbol energy, and $s(i)$ and $a_i(t)$ are the i^{th} symbol and its spreading waveform, respectively. The spreading waveform $a_i(t)$ is composed of a complex sequence of code chips convolved with a pulse shape. It is assumed that the spreading waveform is normalized so that $\int_{-\infty}^{\infty} |a_i(t)|^2 dt = 1$.

If the multi-path channel is discrete and comprises L rays, then

$$15 \quad g(t) = \sum_{l=1}^L g_l \delta(t - \tau_l) \quad (5)$$

represents its impulse response where g_l and τ_l are the complex gain coefficient and delay of the l^{th} ray, respectively. The signal received through this channel may be represented by:

$$r(t) = \sqrt{E} \sum_{l=1}^L \sum_{i=-\infty}^{\infty} g_l s(i) a_i(t - iT - \tau_l) + n(t), \quad (6)$$

where $n(t)$ is a complex noise process.

20 The finger of a RAKE receiver incorporates a delay, d , which precedes a correlator matched to the spreading waveform of the m^{th} symbol, $a_m(t)$. The output of the finger may be given as:

$$y(d, m) = \int_{-\infty}^{\infty} r(\alpha + d) a_m^*(\alpha) d\alpha$$

$$= \sqrt{E} \int_{-\infty}^{\infty} \sum_{l=1}^L \sum_{i=-\infty}^{\infty} g_l s(i) a_i(\alpha + d - iT - \tau_l) a_m^*(\alpha) d\alpha + \tilde{n}(d, m), \quad (7)$$

where $\tilde{n}(d, m)$ is the complex noise output of the correlator. Assuming that the symbol of interest is $s(0)$, the traditional RAKE would have $m = 0$ and the delays of the fingers would be matched to the delays of the resolvable rays of the multi-path. However, for the M-RAKE
5 200 of the present invention, the spreading waveform used for correlation may be generalized to be any symbol's spreading waveform. Equation 7 may be rewritten as:

$$\begin{aligned} y(d, m) &= \int_{-\infty}^{\infty} r(\alpha + d) a_m^*(\alpha) d\alpha \\ &= \sqrt{E} \int_{-\infty}^{\infty} \sum_{l=1}^L \sum_{i=-\infty}^{\infty} g_l s(i) a_i(\alpha + d - iT - \tau_l) a_m^*(\alpha) d\alpha + \tilde{n}(d, m) \\ &= \sqrt{E} \sum_{i=-\infty}^{\infty} s(i) \sum_{l=1}^L g_l R_{i,m}(d - iT - \tau_l) + \tilde{n}(d, m) \\ 10 &= \sum_{i=-\infty}^{\infty} h_i(d - iT, m) s(i) + \tilde{n}(d, m) \end{aligned} \quad (8)$$

where

$$\tilde{n}(t, m) = \int_{-\infty}^{\infty} n(\alpha + t) a_m^*(\alpha) d\alpha \quad (9)$$

is the colored noise output of the correlator, and

$$R_{i,m}(t) = \int_{-\infty}^{\infty} a_i(\alpha + t) a_m^*(\alpha) d\alpha \quad (10)$$

15 is the correlation of the i^{th} symbols spreading waveform against the m^{th} symbol's spreading waveform, used by the finger's correlator. Finally, the components associated with a particular symbol (a correlation between symbol m and i) may be grouped together into

$$h_i(t, m) = \sum_{l=1}^L g_l R_{i,m}(t - \tau_l), \quad (11)$$

a net channel impulse response for the i^{th} symbol, where spreading code and pulse shape
20 information provides $R_{i,m}(t)$ and channel estimation is used to get g_l .

Finally, the RAKE finger outputs may be vectorized for the detection of the symbol of interest as:

$$\mathbf{y} = [y(d_1, m_1), y(d_2, m_2), \dots, y(d_J, m_J)]^T, \quad (12)$$

where there are J RAKE fingers 220. In a traditional matched filter case, $J = L$, the number of resolvable rays in the multi-path channel. Also, $J > L$ fingers may be used.

The calculation of the noise covariance or correlation matrix \mathbf{R}_n referenced in Equation 3 is computed in one of two ways. In the first way, the noise correlation matrix represents white noise, $n(t)$, which has been colored by the correlation process (filtering) of the RAKE fingers 220 to produce $\tilde{n}(d, m)$, where the noise is a function of the delay d of the RAKE finger 220 and the particular spreading waveform of symbol m used by the correlator 224.

Defining the noise output vector of the RAKE fingers 220 as

$$\mathbf{n} = [\tilde{n}(d_1, m_1) \quad \tilde{n}(d_2, m_2) \quad \dots \quad \tilde{n}(d_J, m_J)]^T, \quad (13)$$

the noise correlation matrix may be represented by:

$$\mathbf{R}_n = E\{\mathbf{n}\mathbf{n}^H\}. \quad (14)$$

The cross-correlation of the i^{th} finger and the j^{th} finger may be given as:

$$\begin{aligned} \mathbf{R}_n(i, j) &= E\{\tilde{n}(d_i, m_i) \tilde{n}^*(d_j, m_j)\} \\ &= E\left\{ \int_{-\infty}^{\infty} n(\alpha + d_i) a_{m_i}^*(\alpha) d\alpha \int_{-\infty}^{\infty} n^*(\beta + d_j) a_{m_j}(\beta) d\beta \right\} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_{m_i}^*(\alpha) a_{m_j}(\beta) E\{n(\alpha + d_i) n^*(\beta + d_j)\} d\alpha d\beta. \end{aligned} \quad (15)$$

Because the noise is white, $E\{n(t) n^*(t + \tau)\} = \sigma^2 \delta(t - \tau)$, where σ^2 is the noise power, Equation 15 becomes:

$$\mathbf{R}_n(i, j) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_{m_i}^*(\alpha) a_{m_j}(\beta) \sigma^2 \delta(\alpha + d_i - \beta - d_j) d\alpha d\beta. \quad (16)$$

Sifting property of the $\delta(\cdot)$ function in integration results in:

$$\begin{aligned}\mathbf{R}_n(i, j) &= \sigma^2 \int_{-\infty}^{\infty} a_0^*(\alpha) a_0(\alpha + d_i - d_j) d\alpha \\ &= \sigma^2 R_{m_j, m_i}(d_i - d_j).\end{aligned}\tag{17}$$

Thus, the noise covariance can be constructed using spreading code cross-correlation information and noise power estimates (σ^2). The latter can be estimated by using known techniques. When interference from other user signals is present (i.e., not modeled as ISI or ICI), then the noise at the receive antenna may not be white. A second way to compute the noise covariance matrix is to estimate noise correlations using despread values. For example, the model in Equation 1 also applies to the pilot symbols being sent on the pilot channel. This model can be used to obtain vectors of noise samples, \mathbf{n} , by subtracting the pilot symbol contributions from the \mathbf{y} vector. Outer products of the noise sample vectors can be formed and smoothed in time to give an estimate of the noise covariance matrix. As this matrix is Hermitian symmetric, only the upper or lower triangle of the matrix needs to be estimated.

Turning now to Figure 3, an alternate embodiment of the present invention, referred to herein as a restricted M-RAKE 300, will now be discussed. Restricted M-RAKE 300 comprises a plurality of RAKE fingers 320 and a multi-channel linear transversal filter (MC-LTF) 340. In restricted M-RAKE 300, each RAKE finger 320 includes a delay element 322 and a correlator 324. In this embodiment, the delays associated with each RAKE finger 320 are chosen so that the despread symbols output from each correlator 324 are time aligned. The despread symbols output from RAKE fingers 320 are input to the MC-LTF 340. As shown in Figure 4, the MC-LTF 340 comprises a plurality of parallel linear transversal filters (LTFs) 342, each associated with a corresponding RAKE finger 320, and a summer 344 that adds the filtered outputs of each LTF 342 to generate a symbol estimate $\hat{s}(m)$. Those skilled in the art will appreciate that other embodiments of the MC-LTF 340 are possible. In

general, past finger outputs are stored and results from multiple fingers and multiple symbol periods are combined.

Each LTF 342 combines despread symbols output from a corresponding RAKE finger 320 during a plurality of successive symbol periods as shown in Figure 5. The despread symbols combined by each LTF 342 include a symbol of interest and the nearby interfering symbols (both preceding and succeeding the symbol of interest) that contribute to ISI. The function of the LTFs 342 is to reduce the ISI in the symbol of interest as described in greater detail below. The filtered signals output from each LTF 342, referred to herein as the filtered output signals, are added by multi-path summer 344 to generate the symbol estimates. The summer 344 performs a similar function as a summer in a conventional RAKE receiver to improve the signal to noise ratio of the final symbol estimates.

Each LTF 342 comprises $2M+1$ weighting elements 346, $2M$ delay elements 348, and LTF summer 350, where M represents the length of the ISI in symbol periods. Thus, when M equals 2, there will be four interfering symbols (two preceding and two succeeding) surrounding a symbol of interest $s(m)$. LTF 342 receives and delays the despread symbols output from a corresponding RAKE finger 320 in a tapped delay line comprising a plurality of delay elements 348. The despread symbols may be corrupted by ISI. During each symbol period, the LTF 342 weights and combines the despread symbols output during $2M + 1$ symbol periods centered on the symbol of interest $s(m)$ in weighting elements 346 and LTF summer 350. The weighting factor for each weighting element 346 is solved for jointly in a single step by processor 330 based on the channel coefficients and the cross-correlation between spreading codes, as described further below.

Restricted M-RAKE 300 reduces the number of RAKE fingers 320 while providing the same RAKE finger outputs typically provided by a larger number of RAKE fingers 220 in the M-RAKE 200 shown in Fig. 2. For example, assume a restricted M-RAKE 300

comprises four RAKE fingers 320, where each RAKE finger output is fed into a multi-channel LTF 340 spanning ± 1 symbols ($M = 1$) around the desired symbol. This exemplary M-RAKE 300 is equivalent to a non-restricted M-RAKE 200 with 12 RAKE fingers 220. Thus processing advantages may be accomplished with restricted M-RAKE 300.

5 Figure 6 shows a linear equalizer (LEQ) 400 constructed in accordance with the present invention. LEQ 400 comprises a generalized RAKE (G-RAKE) 410 and a linear transversal filter 442 to combine successive outputs from the G-RAKE 410. Those skilled in the art will understand that the G-RAKE 410 may use a variety of combining methods, including traditional RAKE combining. G-RAKE 410 comprises a plurality of RAKE
10 fingers 420, a RAKE processor 430, and a RAKE combiner 432. Each RAKE finger 420 comprises a delay element 422 and a correlator 424. Like the restricted M-RAKE 300 of Figure 3, the delays associated with each RAKE finger 420 are chosen so that the despread symbols output from each correlator 424 are time aligned. The outputs of correlators 424 are weighted in weighting elements 426 and summed by summer 434 to form a RAKE output
15 signal. The RAKE output signal from combiner 432 is input to LTF 442. As with the restricted M-RAKE 300 discussed above, the delay length of the LTF 442 is $2M + 1$ symbols centered about the symbol of interest $s(m)$. During each symbol period, LTF 442 combines $2M + 1$ RAKE output signals to generate a symbol estimate. The RAKE output signals are combined using weighting factors determined based on channel coefficients and cross-
20 correlations between spreading codes to reduce ISI from the symbol of interest $s(m)$.

The weighting factors for the LTF 442 are determined according to essentially the same ML criteria way as with the M-RAKE 200, with some redefining of the data vectors in question. As the RAKE finger outputs are stored in LTF 442, the vector of RAKE finger outputs are redefined according to the time they were produced. That is, the outputs of
25 correlators 424, previously defined in Equation 1, may be rewritten as:

$$\begin{aligned} \mathbf{y}(mT) &= [y(mT + d_1, m) \ y(mT + d_2, m) \ \dots \ y(mT + d_J, m)] \\ &= \mathbf{h}_0(mT)s(0) + \sum_{i \neq 0} \mathbf{h}_i(mT)s(i) + \mathbf{n}(mT), \end{aligned} \quad (18)$$

where $\mathbf{h}_i(mT)$ is the response from the i^{th} symbol seen in the m^{th} RAKE finger output vector, and where the symbol of interest is represented by $s(0)$. The output of combiner 432

5 comprises the m^{th} -combined output of the RAKE finger outputs and may be defined as:

$$z(mT) = \mathbf{g}^H \mathbf{y}(mT), \quad (19)$$

where, for example, the weighting and combining process simply comprises the complex conjugate of the coefficients of the channel taps in the dispersive channel, (traditional RAKE) $\mathbf{g} = [g_1 \ g_2 \ \dots \ g_L]^T$. The contents of the LTF 442 may be represented in vector form as:

$$\begin{aligned} \mathbf{z} &= [z(-MT) \ z(-(M-1)T) \ \dots \ z(MT)]^T \\ &= \mathbf{h}'_0 s(0) + \sum_{i \neq 0} \mathbf{h}'_i s(i) + \mathbf{n}', \end{aligned} \quad (20)$$

Equation 20 has the same general form as Equation 1, with the components of the various symbols and noise separated. The estimate $\hat{s}(m)$ of the symbol of interest may be represented as:

$$15 \quad \hat{s}(0) = \mathbf{v}^H \mathbf{z}. \quad (21)$$

The weight vector, \mathbf{v} , in LTF 242 may be calculated by:

$$\mathbf{v} = \mathbf{R}_{u'}^{-1} \mathbf{h}'_0 \quad (22)$$

where the cross-correlation matrix $\mathbf{R}_{u'}$ of the undesired portion of the signal may be given as:

$$\mathbf{R}_{u'} = \sum_{i \neq 0} \mathbf{h}'_i \mathbf{h}'_i{}^H + \mathbf{R}_{n'}. \quad (23)$$

20 The cross-correlation matrix $\mathbf{R}_{u'}$ describes cross-correlations between the effective spreading codes for different symbols. Details regarding the calculation of the noise correlation matrix $\mathbf{R}_{n'}$ are discussed further below. As with the M-RAKE 200 of Figure 2,

the weighting factors calculated according to the ML criteria change each symbol period due to the change in the spreading waveform.

Substituting Equation 18 into Equation 19 produces:

$$\begin{aligned} z(mT) &= \mathbf{g}^H \mathbf{y} \\ &= \mathbf{g}^H \mathbf{h}_0(mT)s(0) + \sum_{i \neq 0} \mathbf{g}^H \mathbf{h}_i(mT)s(i) + \mathbf{g}^H \mathbf{n}(mT), \end{aligned} \quad (24)$$

from which the vector \mathbf{z} may be written as:

$$\mathbf{z} = \begin{bmatrix} z(MT) \\ \vdots \\ z(-MT) \end{bmatrix} = \begin{bmatrix} \mathbf{g}^H \mathbf{h}_0(MT) \\ \vdots \\ \mathbf{g}^H \mathbf{h}_0(-MT) \end{bmatrix} s(0) + \sum_{i \neq 0} \begin{bmatrix} \mathbf{g}^H \mathbf{h}_i(MT) \\ \vdots \\ \mathbf{g}^H \mathbf{h}_i(-MT) \end{bmatrix} s(i) + \begin{bmatrix} \mathbf{g}^H \mathbf{n}(MT) \\ \vdots \\ \mathbf{g}^H \mathbf{n}(-MT) \end{bmatrix} \quad (25)$$

From this \mathbf{h}'_i may be defined as

$$\mathbf{h}'_i = [\mathbf{g}^H \mathbf{h}_i(MT) \quad \mathbf{g}^H \mathbf{h}_i((M-1)T) \quad \dots \quad \mathbf{g}^H \mathbf{h}_i(-MT)]^T, \quad (26)$$

and the noise vector, \mathbf{n}' , as:

$$\mathbf{n}' = [\mathbf{g}^H \mathbf{n}(MT) \quad \mathbf{g}^H \mathbf{n}((M-1)T) \quad \dots \quad \mathbf{g}^H \mathbf{n}(-MT)]^T. \quad (27)$$

This noise vector \mathbf{n}' may be rewritten as:

$$\mathbf{n}' = \begin{bmatrix} \mathbf{g}^H & \mathbf{0}^H & \dots & \mathbf{0}^H \\ \mathbf{0}^H & \mathbf{g}^H & \dots & \mathbf{0}^H \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^H & \mathbf{0}^H & \dots & \mathbf{g}^H \end{bmatrix} \begin{bmatrix} \mathbf{n}(MT) \\ \vdots \\ \mathbf{n}(-MT) \end{bmatrix} = \mathbf{G}^H \mathbf{n}'' , \quad (28)$$

where this block diagonal matrix of channel coefficients may be defined as:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{g} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{g} \end{bmatrix} \quad (29)$$

This allows the correlation matrix $\mathbf{R}_{n'}$ of the noise vector, \mathbf{n}' , to be written as

$$\begin{aligned}\mathbf{R}_{n'} &= E\{\mathbf{n}'\mathbf{n}'^H\} \\ &= \mathbf{G}^H E\{\mathbf{n}''\mathbf{n}''^H\}\mathbf{G} \\ &= \mathbf{G}^H \mathbf{R}_{n''} \mathbf{G},\end{aligned}\tag{30}$$

Equation 30 shows that $\mathbf{R}_{n''}$ may be calculated element by element in the same manner in which \mathbf{R}_n was calculated (e.g., Equation 17 for the white noise case).

5 The present invention may also be utilized in multi-code receivers. Multi-code transmission results when multiple code channels are devoted to a single downlink receiver to increase data carrying capability. This is analogous to several separate simultaneous transmissions being received by a single receiver. As the receiver is bound to demodulate each of these code channels, it can use the results of the demodulation of one code channel to
10 aid in the demodulation of another code channel. Further, because of the multi-codes received by the multi-code receiver, each demodulated symbol may, in addition to ISI, be susceptible to inter-code interference (ICI) caused by symbols from adjacent codes.

For example, the M-RAKE 200 of Figure 2 may be adapted to process multi-code signals by assigning a subgroup of RAKE fingers 220 to each code channel. While those
15 skilled in the art will understand that tuning to the spreading codes of other code channels does not necessarily demodulate the other code channels, they will also appreciate that this tuning process provides information on the other code channels, which may help to reduce cross interference effects on the code channel of interest. In its most general form, the modified M-RAKE includes multiple subgroups of RAKE fingers 220 tuned to the spreading
20 codes of different code channels. Further, these RAKE fingers 220 may be tuned to different symbol intervals and delays.

The restricted M-RAKE 300 of Figure 3 may also be extended to multi-code by adding additional RAKE fingers 320 tuned to different code channels, with expansion of the

MC-LTF 340 according to the number of added RAKE fingers 320. In this scenario, the joint solution of the weights within the MC-LTF 340 takes into account the spreading codes of all code channels.

Similarly, the LEQ 400 of Figure 6 may also be extended to multi-code by using a plurality of LEQs 400 tuned to different code channels and expanding the LTF 442 following each LEQ 400 to an MCO-LTF as shown in Figure 7. The multi-code linear equalizer (MCO-LEQ) 500 comprises three G-RAKEs 510, each tuned to one of three code channels, a RAKE processor 530, and a multi-code linear transversal filter 540. Each G-RAKE 510 comprises a plurality of RAKE fingers 520 and a RAKE combiner 532. Each RAKE finger includes a delay element 522 and a correlator 524. The delays for the delay element 522 are chosen such that the output from each RAKE finger 520 is time aligned with the other RAKE fingers 520. Correlators 524 despread the received symbols, which are then weighted in weighting elements 526 and summed by summer 534. The weights applied by the weighting elements 526 are calculated in a conventional RAKE or G-RAKE manner as is well known in the art. The MCO-LTF 540 combines the RAKE output symbols from each G-RAKE 510 using weighting factors determined based on cross-correlations between symbols based on channel coefficients and cross-correlations between spreading codes as previously described. In the situation where all of the code channels are allocated to a single user, the MCO-LTF 540 outputs one symbol estimate for each code channel (co-channel). Thus, if there are three code channels, the MCO-LTF 540 would output three symbols per symbol period. In this case, the MCO-LTF 540 would reduce intersymbol interference attributable to symbols transmitted on the same code channel, as well as intercode interference due to symbols transmitted over different co-channels. The present invention, however, may also be used in the situation where each co-channel is allocated to a different user.

Figure 8 illustrates an exemplary MCO-LTF 540 corresponding to MC-LEQ 500 of Figure 7. The MCO-LTF 540 may, in one embodiment of the invention, comprise a plurality of multi-channel linear transversal filters (MC-LTF) 340 as shown in Figure 4. Each MC-LTF 340 combines RAKE output symbols from all of the G-RAKEs 510 using weighting factors determined based on channel coefficients and cross-correlations between symbols. Each MC-LTF 340 in the MCO-LTF 540 outputs a single symbol estimate corresponding to a selected one of the co-channels for each symbol period. The weighting factors for each LTF 342 would differ. For example, the weighting factors for the MC-LTF 340 associated with co-channel 1 would be selected to reduce interference from the symbols on co-channels 2 and 3. The same approach would be used for the MC-LTF 340 for co-channels 2 and 3, resulting in different weighting factors. Note that the LTF units that store the same RAKE outputs may share memory to reduce memory requirements.

As discussed above, the weighting factors for LTF 342 may be determined according to an ML solution. However, computational savings may be achieved if the ML solution is calculated as a minimum mean squared error (MMSE) solution, and then converted to a ML solution through a scaling of the combining weights (or final output), which can be calculated from elements of the MMSE solution.

It is straightforward to show that for a single user detection RAKE receiver 200, 300, 400 the weight vector, \mathbf{w} , ML combining of the RAKE finger outputs is the same as the MMSE weight vector solution, \mathbf{v} , within a scaling factor. That is,

$$\mathbf{w} = a\mathbf{v} \quad (31)$$

where a is some real coefficient. This a coefficient essentially relates the reliability of the symbol estimate.

Examining first the single user case, the RAKE finger outputs may be written as:

$$\mathbf{y} = \mathbf{h}_0 s(0) + \sum_{i \neq 0} \mathbf{h}_i s(i) + \mathbf{n}, \quad (32)$$

where \mathbf{h}_i is the net response across the fingers of the i^{th} symbol, $s(i)$, through spreading, channel, and despreading. The noise vector \mathbf{n} represents the vectorized output of multiple fingers with white noise as the input. Without loss of generality, $\mathbf{h}_0 s(0)$ corresponds to the symbol of interest where $m=0$.

- 5 For the MMSE solution, the data correlation matrix, $\mathbf{R}_y = E\{\mathbf{y}\mathbf{y}^H\}$, may be used to solve for the familiar expression

$$\mathbf{v} = \mathbf{R}_y^{-1} \mathbf{h}_0, \quad (33)$$

where \mathbf{h}_0 results from $E\{\mathbf{y}s^*(0)\}$, with this expectation taken only over the symbols.

- For the ML solution, the disturbance correlation matrix, \mathbf{R}_u , is used, which is the data
10 correlation matrix less the part related to the desired symbol, or

$$\mathbf{R}_u = \mathbf{R}_y - \mathbf{h}_0 \mathbf{h}_0^H, \quad (34)$$

to solve for another similar, yet familiar expression

$$\mathbf{w} = \mathbf{R}_u^{-1} \mathbf{h}_0. \quad (35)$$

- Equation 31 may be proven easily, starting with Equation 33 and using the fact of Equation
15 34:

$$\begin{aligned} \mathbf{v} &= \mathbf{R}_y^{-1} \mathbf{h}_0 \\ &= \mathbf{R}_u^{-1} \mathbf{R}_u \mathbf{R}_y^{-1} \mathbf{h}_0 \\ &= \mathbf{R}_u^{-1} [\mathbf{R}_y - \mathbf{h}_0 \mathbf{h}_0^H] \mathbf{R}_y^{-1} \mathbf{h}_0 \\ &= [\mathbf{R}_u^{-1} \mathbf{h}_0] - [\mathbf{R}_u^{-1} \mathbf{h}_0] \cdot \mathbf{h}_0^H [\mathbf{R}_y^{-1} \mathbf{h}_0] \\ &= \mathbf{w} - \mathbf{w} \mathbf{h}_0^H \mathbf{v}. \end{aligned} \quad (36)$$

Because $\mathbf{h}_0^H \mathbf{v}$ is a scalar, \mathbf{w} may be written as:

$$\mathbf{w} = \frac{\mathbf{v}}{1 - \mathbf{h}_0^H \mathbf{v}}, \quad (37)$$

resulting in a scaling factor of:

$$a = \frac{1}{1 - \mathbf{h}_0^H \mathbf{v}}. \quad (38)$$

This scaling factor, a , is written in terms of an MMSE solution, with the assumption that it is desirable to convert an MMSE solution to an ML solution. That is, the terms for calculating a would already have been calculated for the MMSE solution.

Other expressions for “ a ” can be used which account for the fact that the channel estimate may be a scaled version of the true channel coefficient, due to differences between the pilot and traffic channel power levels and spreading factors. Specifically, the factor “ a ” can be computed as

$$a = \frac{\sqrt{A}}{1 - A \mathbf{h}_0^H \mathbf{v}} \quad (39)$$

where A corresponds to the ratio of the energy in a traffic symbol to the energy in the pilot symbol. For example, A may be expressed as

$$A = r_1 r_2, \quad (40)$$

where r_1 is the ratio of power allocated to the traffic channel to the power allocated to the pilot channel. This ratio can be estimated or set to some nominal value. Estimation of this ratio is addressed in Bottomley et al., “Communications methods, apparatus, and computer program products using gain multipliers,” U.S. Patent Application No. 09/968,443, filed 1 October 2001. Variable r_2 represents the ratio of the spreading factor of the traffic channel to the spreading factor of the pilot channel (length of the pilot symbol used for channel estimation). These quantities are known at the receiver.

Regarding the multi-user/multi-code detection case, which may be used in the WCDMA multi-code transmission, parallel data channels sent to one user are considered as multiple user signals. Separate banks of RAKE fingers, each tuned to a user spreading sequence, can produce a vector of RAKE finger outputs, \mathbf{z} , and these may be combined

together to a *vector* symbol estimate, where the elements of the vector are the estimates of the individual user symbols. For the MMSE case, this may be represented by $\hat{\mathbf{s}}_{MMSE} = \mathbf{V}^H \mathbf{z}$ and for the ML case, $\hat{\mathbf{s}}_{MLSE} = \mathbf{W}^H \mathbf{z}$ where \mathbf{V} and \mathbf{W} are the MMSE and ML weight matrices, respectively. The same scaled relationship between the MMSE multi-user weight solution and the ML multi-user weight solution is not as obvious as in the single user case.

Now consider the multi-code/multi-user case, where there are K codes in parallel. The RAKE finger output vector may be given by:

$$\mathbf{z} = \sum_{i \neq 0} \sum_{k=1}^K \mathbf{h}_{i,k} s(i, k) + \mathbf{n} \quad (41)$$

where $\mathbf{h}_{i,k}$ is the net response of the k^{th} user's i^{th} symbol, $s(i, k)$. Again, \mathbf{n} is the noise vector output from the fingers.

Solving jointly for a vector of user symbols

$$\mathbf{s}(i) = [s(i,1) \ s(i,2) \ \dots \ s(i,K)]^T \quad (42)$$

produces:

$$\begin{aligned} \hat{\mathbf{s}}_{MMSE}(0) &= \begin{bmatrix} \mathbf{v}_1^H \mathbf{z} \\ \mathbf{v}_2^H \mathbf{z} \\ \vdots \\ \mathbf{v}_K^H \mathbf{z} \end{bmatrix} \\ &= [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_K]^H \mathbf{z}, \end{aligned} \quad (43)$$

where \mathbf{v}_k is weight vector to provide the MMSE estimate of the k^{th} user's symbol. This may be rewritten as a matrix \mathbf{V} times the RAKE finger outputs

$$\hat{\mathbf{s}}_{MMSE}(0) = \mathbf{V}^H \mathbf{z}, \quad (44)$$

where $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_K]$ and $\hat{\mathbf{s}}_{MMSE}(0)$ represents the symbol of interest. It's straightforward to show that the joint MMSE solution that minimizes $\|\mathbf{s}(0) - \hat{\mathbf{s}}(0)\|^2$ is the same as the solution that minimizes the squared error of the individual estimates of the user's symbols,

$|s(0,k) - \hat{s}(0,k)|^2$. As a result, the weight matrix for the joint MMSE solution, \mathbf{V} , may be written as:

$$\begin{aligned} \mathbf{V} &= [\mathbf{R}_y^{-1} \mathbf{h}_{0,1} \quad \mathbf{R}_y^{-1} \mathbf{h}_{0,2} \quad \dots \quad \mathbf{R}_y^{-1} \mathbf{h}_{0,k}] \\ &= [\mathbf{R}_y^{-1} \mathbf{h}_{0,1} \quad \mathbf{h}_{0,2} \quad \dots \quad \mathbf{h}_{0,k}] \\ &= \mathbf{R}_y^{-1} \mathbf{H}_0, \end{aligned} \tag{45}$$

where the net responses of the K users become the K columns of the matrix \mathbf{H}_0 , the 0 subscript denotes that this is the response for the desired 0^{th} symbol vector. Going forward, the 0 may be dropped for clarity, understanding that the response vectors for the 0^{th} symbol are being considered.

10 All of this vector and matrix manipulation provides a concise way of writing a joint MMSE solution. Assuming that this equation is solvable using matrix inversion, the implication of Equation 45 is that only a single matrix inversion needs to be calculated. For the ML solution, it is not obvious that this is the case. The vector ML solution may be written in much the same way as before

$$15 \quad \hat{\mathbf{s}}_{ML}(0) = \mathbf{W}^H \mathbf{z} \tag{46}$$

where $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_K]$. As a solution that gives us log likelihood ratios (LLR) for the bits of the individual users is desired, the estimates of the symbols to be ML individually is desired.

The weight matrix, \mathbf{W} , for the ML solution may therefore be represented by:

$$20 \quad \mathbf{W} = [\mathbf{R}_{u,1}^{-1} \mathbf{h}_1 \quad \mathbf{R}_{u,2}^{-1} \mathbf{h}_2 \quad \dots \quad \mathbf{R}_{u,K}^{-1} \mathbf{h}_K], \tag{47}$$

where the disturbance correlation matrix for each user's ML weight vector solution may be different, as denoted $\mathbf{R}_{u,k}$ for the k^{th} user. Keeping the assumption that the weights using matrix inversion are being solved for, it would appear that it is necessary to calculate K

different inverses to solve this system, as opposed to the single inverse matrix of the MMSE solution.

However, it should be noted that as in the single user case, the disturbance correlation matrix for the ML solution is the same as the MMSE data correlation matrix less the component from the desired symbol, that is

$$\mathbf{R}_{u,k} = \mathbf{R}_y - \mathbf{h}_k \mathbf{h}_k^H. \quad (48)$$

It is recognized that the inverse of $\mathbf{R}_{u,k}$ can be written as a “rank one update” inverse so,

$$\mathbf{R}_{u,k}^{-1} = \mathbf{R}_y^{-1} + \frac{\mathbf{R}_y^{-1} \mathbf{h}_k \mathbf{h}_k^H \mathbf{R}_y^{-1}}{1 - \mathbf{h}_k^H \mathbf{R}_y^{-1} \mathbf{h}_k}, \quad (49)$$

The k^{th} column of \mathbf{W} becomes:

$$\begin{aligned} \mathbf{w}_k &= \mathbf{R}_y^{-1} \mathbf{h}_k + \frac{\mathbf{R}_y^{-1} \mathbf{h}_k \mathbf{h}_k^H \mathbf{R}_y^{-1} \mathbf{h}_k}{1 - \mathbf{h}_k^H \mathbf{R}_y^{-1} \mathbf{h}_k} \\ &= \mathbf{v}_k + \frac{\mathbf{v}_k \mathbf{h}_k^H \mathbf{v}_k}{1 - \mathbf{h}_k^H \mathbf{v}_k} \\ &= \mathbf{v}_k \left(1 - \frac{\mathbf{h}_k^H \mathbf{v}_k}{1 - \mathbf{h}_k^H \mathbf{v}_k} \right) \\ &= \frac{\mathbf{v}_k}{1 - \mathbf{h}_k^H \mathbf{v}_k}, \end{aligned} \quad (50)$$

and the same conclusion as for the single user case is achieved, although shown from a slightly different direction (using the rank one update). Then Equation 47 becomes:

$$\mathbf{W} = [\lambda_1 \mathbf{v}_1 \quad \lambda_2 \mathbf{v}_2 \quad \dots \quad \lambda_K \mathbf{v}_K] \quad (51)$$

where

$$\lambda_k = \frac{1}{1 - \mathbf{h}_k^H \mathbf{v}_k}, \quad (52)$$

and ultimately where:

$$\mathbf{W} = \mathbf{V}\mathbf{\Lambda} \quad (53)$$

where $\mathbf{\Lambda}$ is a diagonal matrix with λ_k as the k^{th} diagonal element.

Grouping these individual solutions in a matrix equation provides a concise expression in Equation 53, but it also points out important implementation points.

- 5 Specifically, given a solution involving matrix inversion, it emphasizes that only a single inverse matrix is calculated for an MMSE solution as opposed to K inverse matrices for the ML solution, which may represent significant processing savings.

As discussed above and shown in Equation 54, the form of the MMSE solution is similar to the form of the ML solution.

$$10 \quad \mathbf{v}_{MMSE} = (\mathbf{R}_z)^{-1} \mathbf{h}'_o \quad (54)$$

However, the correlation matrix of the data, \mathbf{R}_z is used, rather than the correlation matrix of the disturbance or noise, \mathbf{R}_u .

The form of the \mathbf{z} vector of RAKE outputs for the q^{th} symbol may be represented by:

$$\mathbf{z}_q = [z(q-M) \ z(q-(M-1)) \ \dots \ z(q+M)]^T, \quad (55)$$

- 15 where the symbol period, T , is dropped for convenience. This vector essentially represents the contents of the LTF 442 corresponding to the LEQ 400 discussed above. In another form, \mathbf{z}_q may be rewritten as:

$$\mathbf{z}_q = [z(q-M) \ \mathbf{a}_q]^T, \quad (56)$$

where most of the \mathbf{z} vector is now represented by subvector \mathbf{a}_q . Note that for the next symbol,

- 20 $q+1$, the data is shifted through LTF 442, yielding:

$$\mathbf{z}_{q+1} = [\mathbf{a}_q \ z(q+M+1)]^T, \quad (57)$$

where $z(q+M+1)$ comprises the new combined RAKE output introduced into the filter.

As discussed above, the correlation matrix for the solution for the q^{th} symbol may be represented by:

$$\mathbf{R}_z(q) = E\{\mathbf{z}_q \mathbf{z}_q^H\}, \quad (58)$$

where $E\{\cdot\}$ is the statistical expectation operation, which in this case is taken only over the random noise. Substituting Equation 56 into Equation 58 results in:

$$\mathbf{R}_z(q) = E \begin{bmatrix} z(q-M)z^*(q-M) & z(q-M)\mathbf{a}_q^H \\ \mathbf{a}_q z^*(q-M) & \mathbf{a}_q \mathbf{a}_q^H \end{bmatrix}. \quad (59)$$

5 Repeating this with Equation 57 for $q+1$ results in:

$$\mathbf{R}_z(q+1) = E \begin{bmatrix} \mathbf{a}_q \mathbf{a}_q^H & \mathbf{a}_q z(q+M+1) \\ z^*(q+M+1)\mathbf{a}_q^H & z(q+M+1)z^*(q+M+1) \end{bmatrix}. \quad (60)$$

As shown in Equations 59 and 60, a submatrix $E\{\mathbf{a}_q \mathbf{a}_q^H\}$ appears in the correlation matrix for symbol q and $q+1$, indicating that it may be reused. Essentially, for each successive symbol sub-matrices $E\{\mathbf{a}_q \mathbf{a}_q^H\}$ may be shifted to partially form a new correlation matrix for the new symbol. Figures 9a and 9b illustrate the nature of this matrix shift. As shown in Figure 9a, \mathbf{R}_q 600 comprises row one 610, column one 620, and submatrix $E\{\mathbf{a}_q \mathbf{a}_q^H\}$ 630. Row one 610 and column one 620 border the leftmost and uppermost boundaries of submatrix $E\{\mathbf{a}_q \mathbf{a}_q^H\}$ 630. Further, Figure 9b illustrates that \mathbf{R}_{q+1} 602 comprises row M 612, column M 622, and submatrix $E\{\mathbf{a}_q \mathbf{a}_q^H\}$ 630. \mathbf{R}_{q+1} 602 may be generated from \mathbf{R}_q by shifting submatrix $\mathbf{a}_q \mathbf{a}_q^H$ 630 into the upper left-hand corner of \mathbf{R}_{q+1} 602. As a result, only row M 612 and column M 622 is calculated to complete \mathbf{R}_{q+1} 602. Depending on the size of \mathbf{R} matrix, the computational savings may be substantial.

The MMSE solution allows for the special time varying structure of the data correlation matrix, \mathbf{R}_z . As noted above, the variation of the spreading waveform necessitates the calculation of a new weight vector for each symbol period. This variation of the spreading waveform manifests itself in the changing of \mathbf{h}'_0 and \mathbf{R}_z^{-1} . However, \mathbf{R}_z does not completely change from symbol to symbol, and therefore, a substantial portion of \mathbf{R}_z may be

reused. As a result, only a portion of \mathbf{R}_z needs to be calculated from scratch for each new symbol.

Further, in the multi-code scenario, the same MC-LEQ 500 with the same data contents may be used to solve for all K code channels at a given symbol instance. If the ML
5 solution is used, the \mathbf{R}_u matrix is recalculated for each code channel because \mathbf{R}_u varies from code channel to code channel. However, with the MMSE solution, a single common data correlation matrix, \mathbf{R}_z may be calculated and used for all K solutions.

The RAKE receiver embodiments discussed herein may be disposed in any wireless communication device, such as a base station 700 or mobile terminal 710, as shown in Figure
10 10. As used herein, the term “mobile terminal” may include a cellular radiotelephone with or without a multi-line display; a Personal Communications System (PCS) terminal that may combine a cellular radiotelephone with data processing, facsimile and data communications capabilities; a personal data assistant (PDA) that can include a radiotelephone, pager, Internet/intranet access, Web browser, organizer, calendar, and/or a global positioning system
15 (GPS) receiver; and a conventional laptop and/or palmtop receiver or other appliance that includes a radiotelephone transceiver. Mobile terminals may also be referred to as “pervasive computing” devices.

Base station 700 includes antenna 702 for transmitting symbols to mobile terminal 710. Objects, such as interfering object 720 may cause multiple echoes of the transmitted
20 symbols to arrive at mobile terminal 710 at different times, as described above. The symbols received at antenna 712 of mobile terminal 710 are then processed in RAKE receiver 714. RAKE receiver 714 represents any of the exemplary embodiments described above. Similarly, mobile terminal 710 may transmit symbols along multiple paths to base station 700. RAKE receiver 704 in base station 700 may then process the received symbols
25 according to any of the exemplary embodiments described above.

In discussing exemplary embodiments of the present invention, it should be understood that one or more embodiments of the present invention comprise signal processing methods that may be implemented in hardware using discrete or integrated circuits, in software as stored program instructions, or in some combination thereof. More
5 generally, one or more embodiments of the present invention may be embodied in hardware and/or software (including firmware, resident software, micro-code, etc.), including an application specific integrated circuit (ASIC).

The present invention may, of course, be carried out in other ways than those specifically set forth herein without departing from essential characteristics of the invention.

10 The present embodiments are to be considered in all respects as illustrative and not restrictive, and all changes coming within the meaning and equivalency range of the appended claims are intended to be embraced therein.